JOCURI EDUCATIVE COLECTIVE
JECO

JOCURI PE CALCULATOR

LOGO

JOCURI PE CALCULATOR — o nouă posibilitate de instruire în timpul liber

După părerea unanim acceptată, calculatorul trebuie privit ca un prieten, o unealtă a omului, o sursă de potențare a puterii de creație cu largi orizonturi.

„Calculatorul — spunea Jean Jaques Servan Schreiber — este o unealtă, așa cum a fost cartea după Guttenberg. Ca și cartea, e o trambulină pentru creatori. În fața revoluției electronice nu există decît o alternativă: ori înveți să controlezi tehnologia, ori te lași controlat de ea".

Jocurile pe calculator nu mai sînt o noutate, dar ceea ce dă importanță domeniului, avînd calculatorul ca partener „inteligent" de întrecere, este și faptul că se vor putea trece aproape toate jocurile logice — așa-zise vechi — pe calculator și că există mari posibilități ca o multitudine de jocuri logice noi să fie lansate pe piață mai întîi (sau numai) pe calculator.

Jocul și calculatorul se completează, deci, atît de bine, încît par făcuți unul pentru celălalt. Totuși, ceva intervine și aici. Astfel, în jocul clasic, regulile trebuie cunoscute de toți participanții. Calculatorul are regulile lui ce nu pot fi modificate sau ignorate. Totodată, jucătorul nu mai este un creator, ci un simplu participant, iar calculatorul nu este niciodată un partener

egal, ci un arbitru și un instrument de simulare. Noua dimensiune pe care o introduce calculatorul, bogăția sa, o constituie multitudinea de situații și variante pe care le poate simula.

Dar calculatorul nu se limitează numai la jocuri. El are o gamă largă de posibilități. Dintre acestea, o mare utilizare o are în domeniul instruirii personale în diverse discipline de învățămînt, respectiv de la matematică, fizică, chimie pînă la limbile străine.

RECOOP este interesat să primească orice sugestie și propunere de programe pe calculator.

Publicul, din ce în ce mai pasionat, caracteristicile deosebite ale calculatorului și, mai ales, extraordinara inventivitate în acest domeniu, garantează viitorul unor noi programe pe calculator.

Dr. GH. FEȚEANU

Limbajul LOGO

Limbajul LOGO *și-a demonstrat calitățile de instruire pentru copii, utilizarea lui permițînd familiarizarea cu calculatorul, dezvoltînd în același timp capacitatea intelectuală, vorbirea și scrisul, imaginația, puterea de a anticipa acțiuni, orientarea, etc.*

LOGO *este un limbaj de programare de nivel înalt, care îmbină simplitatea — modul de adresare se apropie de cel natural — cu un larg spectru de facilități, care permit lucrul interactiv, repetarea unor acțiuni, definirea de noi proceduri de lucru, pe baza setului de comenzi existente și utilizarea lor recursivă. Astfel, sub forma unui joc, se pot însuși noțiuni de bază de geometrie, aritmetică, proiectare și desen, trigonometrie, muzică, dar și tehnici avansate de programare a calculatoarelor. Jucîndu-se cu* **LOGO,** *copilul invață, descoperind singur legi și fenomene, intr-un proces de instruire liberă și autocondusă.*

Pentru utilizarea limbajului **LOGO** *este necesară următoarea configurație:*

– **un calculator personal compatibil cu tipul Sinclair Spectrum (CIP, HC, TIM-S, COBRA)** *cu o memorie interna de 48 Ko;*

– **un televizor sau monitor** *alb/negru sau color (acesta din urma fiind de preferat deoarece* **LOGO** *are comenzi speciale pentru lucrul cu culori);*

un **casetofon uzual**

caseta magnetică LOGO, *editata de* **RECOOP** *in colaborare cu* **ITCI.**

Fața 1

LOGO

Pentru a se putea lucra cu limbajul de programare **LOGO**, acesta va trebui incarcat de pe caseta magnetica. Deci, puneți caseta de la inceput, introduceți comanda **LOAD" "** și porniți casetofonul. În momentul în care apare pe ecran mesajul **„BINE AȚI VENIT ÎN LOGO"** și un semn de întrebare, înseamna ca puteți să opriți casetofonul și să incepeți sa introduceți comenzi **LOGO.**

Dar atenție! **LOGO** face distincție între literele mici și cele mari, iar comenzile **LOGO** trebuie scrise cu litere mari, litera cu litera. În consecință, primul lucru pe care il veți face este sa treceți sistemul în modul de scriere cu litere mari, prin acționarea simultană a tastelor **CS** și **2.**

ROMÂNA

Daca doriți să folosiți și **comenzi LOGO în limba română,** va trebui sa le încarcați, în prealabil, în memoria calculatorului, ele constituind urmatorul fișier de pe caseta magnetică. Observați ca în **LOGO** vom folosi în locul termenului de program (ca ansamblu de instrucțiuni memorate de calculator) pe cel de fișier (ca ansamblu de proceduri-secvențe de comenzi **LOGO,** memorate de calculator sub un nume). **Încărcarea** fișierului de comenzi în limba română se face cu comanda **LOAD „ROMANA",** dupa care veți porni casetofonul. În timpul încărcării numele comenzilor în limba româna vor fi afișate pe ecran. Dupa încărcare, va aparea din nou pe ecran semnul întrebării, care semnifică faptul că se așteapta o comanda **LOGO.** Veți putea introduce acum atît comenzile originale, cît și pe cele în limba româna al căror titlu l-ați observat pe ecran.

DESENE

Următoarele fişiere de pe caseta magnetică vor evidenţia cîteva din posibilităţile şi facilităţile oferite de limbajul **LOGO.** Astfel, printre cele mai importante, se numără cele grafice şi muzicale, sugestiv demonstrate prin următorul fişier de pe casetă, intitulat **DESENE. Încărcarea** lui în memoria calculatorului se va face prin comanda **LOAD „DESENE“**, în timp ce **lansarea în execuţie** (după ce apare semnul întrebării pe ecran) prin simpla tastare a cuvîntului **DESENE.**

Veţi observa desenarea pe ecran a cîtorva spirale colorate interesant, realizate prin intermediul tehnicilor de recursivitate pe care limbajul **LOGO** le pune la dispoziţia utilizatorilor.

JOC

Următorul fişier este un joc ce se poate folosi în vederea acomodării cu distanţele de pe ecran care, în **LOGO**, se măsoară în „paşi“. Este recomandat, deci, ca jocul să se practice înainte de învăţarea utilizării comenzilor FORWARD (ÎNAINTE) şi BACK (ÎNAPOI). **Încărcarea** jocului se face prin comanda **LOAD „JOC“**, iar **lansarea în execuţie** (pornirea jocului) prin tastarea cuvîntului **JOC.** Instrucţiunile de utilizare se vor afişa pe ecran.

„Broasca“ (un triunghi în care vîrful ascuţit indică direcţia) poate fi rotită prin intermediul **tastelor S** pentru **stînga** sau **D** pentru **dreapta** pentru a fi orientată spre o ţintă afişată pe ecran sub forma unui pătrat. După ce broasca a fost orientată, jucătorul va trebui să acţioneze tasta **T** şi, apoi, să indice numărul de „paşi“ pe care trebuie să-i efectueze „broasca“ pentru a lovi ţinta. Distanţa parcursă de broască va fi echivalentă cu cea obţinută pe baza comenzii **LOGO** FORWARD (sau ÎNAINTE), urmată de numărul de paşi respectivi. Dacă broasca intră cu vîrful în pătrat, jucătorul va fi felicitat. Jocul poate fi continuat pînă cînd jucătorul va acţiona tastele **CS** şi **SPACE** pentru întreruperea jocului.

ŞAH

Ultimul fişier de pe prima faţă a casetei nu se referă la jocul de şah, fiind numai o frumoasă imagine grafică a unei table de şah cu piese în perspectivă. **Încărcarea** acestei imagini se va face cu comanda **LOADSCR „ŞAH"**, imaginea apărînd pe ecran în timpul încărcării. După încărcare, imaginea se poate modifica sau şterge cu comenzi **LOGO** sau, pur şi simplu, se poate indica o altă comandă **LOGO.**

Faţa 2

LOGO

Primul fişier de pe faţa a 2-a a casetei este identic cu cel de pe faţa 1, fiind repetat ca o masură de siguranţa. Fişierul poate fi utilizat în cazul deteriorării celui de pe faţa 1 sau în cazul în care cel de pe faţa 1 nu se poate incarca din cauza unei erori de citire (apare mesajul „Tape loading error").

Încărcarea se face (la fel) cu comanda **LOAD" ".**

Nu uitaţi, sa treceţi în modul de scriere cu litere mari (**CS + 2**) după încărcare.

TITATO

Este vorba de jocul cunoscut şi sub numele de **X** şi **O** pe care acum le puteţi să îl practicaţi jucînd împotriva calculatorului. **Încărcarea** jocului se face cu comanda **LOAD „TITATO"**, în timp ce **lansarea în execuţie** se face (dupa apariţia semnului de întrebare) prin tastarea **TTT**. Atenţie! Incărcarea jocului **TITATO** se va face fara sa existe în memorie şi setul de comenzi informaţii în limba româna, deoarece în acest caz procedurile pentru joc nu vor mai avea loc în memorie. Dacă, totuşi, exista mai multe proceduri în

memorie, atunci, înainte de încărcarea lui **TITATO**, se va proceda la ştergerea lor cu ajutorul comenzii **ERASE** (sau **UITA** dacă este încărcat setul de comenzi în limba română).

După începerea jocului se trasează pe ecran o grilă de 3 X 3 poziţii şi jucătorul are posibilitatea să aleagă cine va efectua prima mutare: jucătorul (tasta **X**) sau calculatorul (tasta **0**).

Pentru efectuarea mutărilor se vor folosi **tastele numerice**: tasta **1** va indica o mutare pe poziţia din colţul stînga sus, în timp ce tasta **9** va indica o mutare pe poziţia din colţul din dreapta jos. Mutările jucătorului se efectuează prin punerea semnului ales (**X** sau **0**) pe poziţia indicată, iar cele ale calculatorului prin punerea celuilalt semn pe o anumită poziţie aleasă de calculator.

Dacă pentru efectuarea unei mutari se indica alta tasta decît una numerică sau se indica o poziţie deja ocupată, apare mesajul „mutare ilegala". iar mutarea se poate repeta. Mutările vor alterna. Scopul jocului este de a realiza 3 semne pe aceeaşi linie (orizontală, verticală sau diagonală).

Dacă nici calculatorul şi nici jucatorul nu reuşesc acest lucru după umplerea tuturor poziţiilor grilei, atunci se da mesajul „REMIZA" şi jocul se poate relua după apăsarea oricărei taste cu comanda **TTT**. Dacă jucătorul ori calculatorul va realiza punerea a 3 semne pe aceeaşi linie, se va da mesajul respectiv („Ai cîştigat!" sau „Am cîştigat!") şi jocul se va putea relua după acţionarea oricărei taste cu comanda **TTT**.

FOTO

Ca şi pe prima faţă a casetei, apare o frumoasă imagine grafică, de data aceasta însă, a unui aparat de fotografiat. **Încărcarea** imaginii se va face cu comanda: **LOADSCR „FOTO"**. Imaginea va apărea pe ecran în timpul încărcării. După apariţia imaginii, se poate introduce orice comanda **LOGO**.

Caseta magnetica **LOGO** a fost realizată de Ion Diamandi şi Cristian Constantinescu.

UN SUMAR AL LIMBAJULUI LOGO

Notă: acest sumar prezintă cîteva reguli de bază ale versiunii limbajului **LOGO** implementat pe calculatoarele **Sinclair Spectrum şi compatibile,** precum şi o foarte succinta descriere a comenzilor şi operaţiilor pe care limbajul le pune la dispoziţie. Din acest punct de vedere materialul se prezintă ca un memorator, adresîndu-se mai degrabă unor persoane avizate decît unor începatori. Pentru această ultima categorie, în scopul învaţării limbajului **LOGO,** recomandam folosirea, împreună cu materialul de faţă a **GHIDULUI LOGO.**

În cadrul unui capitol, comenzile şi operaţiile sînt descrise în ordine alfabetica.

Se folosesc următoarele notaţii: **CS** pentru **CAPS SHIFT; SS** pentru **SYMBOL SHIFT; CS + tastă** pentru acţionarea simultană a celor două taste.

Manevrarea cursorului

Ştergerea caracterului dinaintea cursorului: **CS + 0**
Deplasare la dreapta: **CS + 8**
Deplasare la stînga: **CS + 5**
Deplasare în jos: **CS + 6**
Deplasare în sus: **CS + 7**

Schimbarea regimului tastaturii şi al ecranului

Regim „l" regim „C": **CS + 2**
(litere mici) (litere mari)
Regim „l" sau „C" regim „E": **CS + SS**
Regim grafic al ecranului regim textual: comanda **TEXTSCREEN (TS).**

Proceduri

Definiţie: se numeşte **procedură** orice secvenţă de instrucţiuni memorate de calculator sub un **nume,** în vederea executării ei la întîlnirea numelui respectiv.

Structura unei proceduri:

TO nume
comenzi

•

•

•

END

Apelarea procedurii: prin numele ei
Întreruperea executării unei proceduri: **CS + SPACE**

Obiecte LOGO

Obiectele LOGO sînt cuvinte sau liste utilizate ca intrări sau ieşiri ale procedurilor.

Un **cuvînt** este o serie de caractere alfabetice sau numerice; pentru a fi deosebite de numele de proceduri, înaintea cuvintelor se pun ghilimele; un cuvînt nu poate conţine caracterul „spaţiu", deoarece acesta indică terminarea cuvîntului. Un cuvînt este, de obicei, delimitat prin spaţii. Mai există şi alţi **delimitatori**, ca de exemplu: [] () = > < + — X / care separă un cuvînt de restul liniei. Între un cuvînt şi oricare din aceşti delimitatori nu este necesară inserarea spaţiului, deşi, uneori, se obişnuieşte acest lucru pentru claritate.

Numerele sînt, de asemenea, cuvinte dar ele se pot scrie fără ghilimele ca şi cuvintele care exprimă valori logice (TRUE — adevărat şi FALSE — fals).

O **listă LOGO** constă dintr-o serie de obiecte **LOGO**, adică cuvinte sau liste; o listă este cuprinsă între paranteze drepte, iar elementele unei liste sînt separate prin spaţii; **lista vidă** (fără nici un element) este [].

Cuvintele pot fi utilizate şi ca **nume de variabile.**

Elementele unui cuvînt sînt caractere, iar elementele unei liste sînt cuvinte sau liste.

Cuvintele sau listele pot fi „depuse" în variabile printr-o instrucţiune de atribuire (MAKE).

Proceduri cu subiect

Pentru a putea funcţiona, unele proceduri necesită subiecte sau parametrii de intrare. Acestea sînt obiecte **LOGO** (cuvinte sau liste). De exemplu:

? PRINT [BUNĂ DIMINEAŢA]

Lista [BUNĂ DIMINEAŢA] este un subiect (sau o intrare) pentru procedura primitivă PRINT. Semnul întrebării este **promptul** specific lui **LOGO** care semnifică faptul că se aşteaptă introducerea unei comenzi **LOGO.**

Comenzi şi operaţii

O procedură poate fi ori o comandă ori o operaţie (funcţie). O operaţie este o procedură care furnizează o valoare, deci,poate fi utilizată ca parametru de intrare pentru altă comandă sau operaţie.

Ghilimele, două puncte, paranteze pătrate şi drepte

LOGO interpretează orice cuvînt ca o procedură. Pentru ca un cuvînt să nu fie interpretat ca o procedură va trebui precedat de unul din următoarele semne:

" ghilimelele indică lui **LOGO** că şirul de caractere care urmează, terminîndu-se cu un spaţiu, este un cuvînt;
: două puncte indică lui **LOGO** că şirul de caractere care urmează, terminîndu-se cu un spaţiu, este numele unui obiect **LOGO.**

[] parantezele pătrate indică lui **LOGO** că elementele dintre aceste paranteze formează o listă.

( ) parantezele rotunde sînt utilizate pentru a permite mai mult de două intrări pentru unele primitive.
De asemenea,parantezele rotunde mai sînt utilizate pentru realizarea unor operaţii aritmetice. De exemplu:

```
? PRINT  2 × (3 + 5)
  16
```

Variabile

În **LOGO** variabilele au un nume. În mod normal, variabilele sînt create cu comanda MAKE (PUNE). De exemplu: ?MAKE "A 1

Valoarea variabilei poate fi obţinută prin punerea caracterului înainte de numele variabilei. De exemplu: ?PRINT:A

O variabilă în **LOGO** poate avea ca valoare o varietate de tipuri de date: o valoare numerică, o listă de valori numerice, un cuvînt, o listă de cuvinte, etc.

O variabilă creată în **LOGO** prin comanda MAKE va fi o variabilă **globală,** adică va exista atît în timpul executarii procedurii, cît și după aceea.

O variabilă declarată ca parametrul de intrare într-o procedură (și nu creată anterior) va fi o variabilă **locală** a acestei proceduri și a oricărei alte proceduri, adică nu va avea atribuită o valoare decît în timpul executării procedurii.

Editorul LOGO

Pentru modificarea procedurilor definite sau chiar pentru definirea noilor proceduri, se folosește **editorul LOGO.**

Apelare: **EDIT (ED)** numele procedurii

Utilizare editor:

CS + 5	deplasează cursorul un caracter la stînga
CS + 6	deplasează cursorul o linie mai jos
CS + 7	deplasează cursorul o linie mai sus
CS + 8	deplasează cursorul un caracter la dreapta
CS + 0	șterge caracterul dinaintea cursorului
CS + 6 (în mod E)	deplasează cursorul la începutul liniei
CS + 6 (în mod E)	deplasează cursorul la sfîrșitul ecranului
CS + 7 (în mod E)	deplasează cursorul la începutul ecranului
CS + 8 (în mod E)	deplasează cursorul la sfîrșitul liniei
B (în mod E)	deplasează cursorul la începutul textului
E (în mod E)	deplasează cursorul la sfîrșitul textului
N (în mod E)	deplasează cursorul la pagina următoare
P (în mod E)	deplasează cursorul la pagina precedentă
Y (în mod E)	șterge de pe ecran linia pe care se află cursorul
R (în mod E)	reinserează linia care a fost ștearsă cu comanda Y (în mod E)

C (în mod E)	părăsirea editorului cu încorporarea modificărilor efectuate și intrarea în **LOGO**
CS + SPACE	părăsirea editorului cu ignorarea modificărilor efectuate și intrarea în **LOGO**

COMENZI ŞI OPERAŢII

1. Broasca ţestoasă

BACK n **BK** n **ÎNAPOI** n **IP** n	deplasare înapoi cu n paşi (pixeli).
BACK GROUND **BG**	furnizează un număr (operaţie) care reprezintă culoarea fondului. Numărul furnizat poate fi: 0 — dacă fondul are culoarea neagră, 1 — albastră, 2 — roşie, 3 — violetă, 4 — verde, 5 — bleu, 6 — galbenă, 7 — albă.

Notă: **broasca ţestoasă** este numele dat micului triunghi care apare pe ecran cînd se utilizează ecranul grafic. Mişcîndu-se dintr-un punct în altul, broasca va lăsa urme, putîndu-se trasa în acest mod linii pe ecran. Ecranul grafic, numit şi cîmpul broaştei ţestoase, apare oricînd se introduce o comandă referitoare la mişcarea broaştei. În mod normal, ecranul grafic are 256 de paşi (puncte) pe orizontală şi 175 de paşi (puncte) pe verticală, iar originea este în centrul ecranului.

CLEAN	ştergerea ecranului fără aducerea broaştei acasă.
CLEARSCREEN **CS** **ŞTERGE**	ştergerea ecranului cu aducerea broaştei acasă.
DOT [x y] **PUNCT** [x y]	desenează un punct la poziţia specificată prin coordonatele x, y. Broasca nu se mişcă de la locul ei şi nu se trasează nici o linie.

FENCE **GARD**	limitează mişcările broaştei în cadrul ecranului grafic.
FORWARD n **FD** n **ÎNAINTE** n **IN** n	deplasare înainte cu n paşi.
HEADING **DIRECŢIE**	furnizează un număr (operaţie) între 0 şi 359 care reprezintă orientarea actuală a broaştei în ecran. 0 = direcţie N, 90 = Est, 180 = Sud, 270 = Vest.
HIDETURTLE **HT** **FĂRĂ BROASCĂ** **FB**	ascunde broasca (face broasca invizibilă).
HOME **ACASĂ**	readucerea broaştei acasă.
LEFT n **LT** n **STÎNGA** n **SA** n	rotire spre stînga cu n° (grade) fără ca broasca să-şi schimbe poziţia.
PENCOLOUR **PC** **CULOARE**	furnizează un număr (operaţie) care reprezintă culoarea cu care va desena broasca. La început culoarea va fi 0 (neagră).
PENDOWN **PD** **CREION** **CR**	peniţa jos: cînd broasca se va mişca ea va trasa o linie.
PENERASE **PE** **GUMA** **GU**	ştergerea liniei: broasca va şterge orice linie peste care va trece. Comenzile PENDOWN, PENUP sau PENREVERSE vor anula efectul comenzii PENERASE.

PENREVERSE **PX** **CI**	peniță inversă: broasca în mișcare va trasa linii, dar va șterge liniile peste care va trece (desenează cu culoarea inversă).
PENUP **PU** **FĂRĂ CREION** **FC**	penița sus: broasca în mișcare nu va trasa linii.
POSITION **POS** **POZIȚIE**	furnizează o listă de două numere (operație) care reprezintă poziția actuală a broaștei.
RIGHT n **RT** n **DREAPTA** n **DR** n	rotire spre dreapta cu n° (grade) fără ca broasca să-și schimbe poziția.
SCRUNCH **RELXY**	furnizează o listă de două numere (operație), care reprezintă proporția dintre mărimea unui pas pe verticală și a unuia pe orizontală.
SETBG n **FIXFOND** n	fixează culoarea fondului (0 — negru, 1 — albastru etc).
SETBORDER n **SETBR** n **FIXCHENAR** n	fixează culoarea chenarului (borderului) (0 — negru, 1 — albastru, etc).
SETHEADING n **SETH** n **FIXDIR** n	rotește broasca, astfel încît orientarea ei să fie n. La început orientarea broaștei este 0.
SETPC n **FIXCULOARE** n	fixează culoarea cu care se va desena (n = 0 culoarea neagră, n = 1 culoarea albastră, etc.)
SETPOS [x y] **FIXPOZ** [x y]	deplasează broasca în punctul de coordonate x și y, fără a-i schimba orientarea (efectuează o translație pînă în punctul respectiv).

SETSCRUNCH [x y] **SETSCR [x y]** **FIXSCARA [x y]**	fixează proporţia dintre mărimea pasului pe verticală şi mărimea pasului pe orizontală. În mod normal (la început) această proporţie este [100 100].
SETX n **FIXX** n	deplasează broasca într-un punct a cărui coordonată pe orizontală (X) este n.
SETY n **FIXY** n	deplasează broasca într-un punct a cărui coordonată pe verticală (Y) este n.
SHOWNP	operaţie care furnizează cuvîntul TRUE dacă broasca este vizibilă (este în modul SHOWTURTLE) sau FALSE dacă nu este.
SHOWTURTLE **ST** **BROASCA**	arată broasca (face broasca vizibilă).
TOWARDS [x y] **CAP [x y]**	furnizează orientarea pe care trebuie să o capete broasca pentru a fi îndreptată spre punctul de coordonate x şi y (operaţie); nu modifică orientarea broaştei.
WINDOW **FEREASTRA**	permite broaştei să se deplaseze în afara ecranului grafic. Cînd broasca este în modul WINDOW, va continua să execute comenzile de deplasare chiar fără a fi văzută.
WRAP **IESEREVINE**	permite broaştei ca atunci cînd iese din ecran să apară din partea opusă a ecranului.
XCOR	furnizează coordonata pe orizontală (abscisa) actuală a broaştei (operaţie).

YCOR	furnizează coordonata pe verticala (ordonata) actuală a broaştei (operaţie).

2. **Cuvinte şi liste**

ASCII caracter	operaţie care furnizează codul ASCII al unui caracter.
BUTFIRST obiect **BF** obiect **FARAPRIMUL** obiect **FP** obiect	operaţie care furnizează obiectul fără primul element al obiectului.
BUTLAST obiect **BL** obiect **FARAULTIMUL** obiect **FU** obiect	operaţie care furnizează obiectul fără ultimul element al obiectului.
CHAR n **CARACTER** n	operaţia care furnizează caracterul al cărui cod ASCII este n. Codul n trebuie să fie un întreg cuprins între 32 şi 165.
COUNT obiect **NREL** obiect	operaţie care furnizează numărul de elemente al obiectului (cuvînt sau listă).
EMPTYP obiect **GOL** obiect	operaţie care furnizează valoarea logică TRUE dacă obiectul **LOGO** este gol şi valoarea logică FALSE în caz contrar.
EQUALP obiect 1, obiect 2 **EGAL** obiect 1, obiect 2	operaţie care furnizează valoarea logică TRUE dacă obiect 1 şi obiect 2 sînt două numere egale, două cuvinte sau liste identice.
FIRST obiect **PRIMUL** obiect	operaţie care furnizează primul element al obiectului. Acesta va fi un caracter dacă obiectul este un cuvînt ori un cuvînt sau o listă dacă obiectul este o listă.

FPUT obiect listă **PUNEPRIMUL** obiect listă	Operație care furnizează o nouă listă, formată prin punerea (alipirea) obiectului la începutul listei (FirstPUT).
ITEM n obiect	Operație care furnizează al n-lea element al unui obiect
LAST obiect **ULTIMUL** obiect	Operație care furnizează ultimul element al obiectului. Acesta va fi un caracter dacă obiectul este un cuvînt sau o listă dacă obiectul este o listă.
LIST obiect 1, obiect 2 (**LIST** obiect 1, obiect 2 ... obiect n) **LISTA** obiect 1, obiect 2 (**LISTA** obiect 1, obiect 2 ... obiect n)	Operație care furnizează o listă ale cărei elemente sînt obiect 1, obiect 2, etc.
LISTP obiect	Operație care furnizează valoarea logică TRUE dacă obiectul este o listă și FALSE în caz contrar. O listă vidă este considerată un cuvînt gol.
LPUT obiect listă **PUNEULTIMUL** obiect listă	Operație care furnizează o nouă listă, formată prin punerea (alipirea) obiectului la sfîrșitul listei (LastPUT).
MEMBERP obiect listă	Operație care furnizează valoarea logică TRUE dacă obiectul este un element al listei și FALSE în caz contrar.
NUMBERP obiect	Operație care furnizează valoarea logică TRUE dacă obiectul este un număr și FALSE în caz contrar.

SENTENCE obiect 1, obiect 2 (**SE** obiect 1 obiect 2 ... obiect n)	Operaţie cu care se formează o listă compusă din obiectele date.
FRAZA obiect 1, obiect 2 (**FRAZA** obiect 1, obiect 2 ... obiect n)	
WORD cuvînt 1, cuvînt 2 (**WORD** cuvînt 1, cuvînt 2 ... cuvînt n)	Operaţie cu care se formează un cuvînt din obiectele date.
WORDP obiect	Operaţie care furnizează valoarea logică TRUE dacă obiectul este un cuvînt şi FALSE în caz contrar.

3. Variabile

MAKE nume obiect **PUNE** nume obiect	atribuie valoarea „obiect“ variabilei „nume“ (pune în locaţia de memorie numită „nume“ valoarea „obiect“).
NAMEP obiect	operaţie care furnizează valoarea logică TRUE dacă obiectul are o valoare şi FALSE în caz contrar.
THING nume	operaţie care furnizează conţinutul unui nume.

4. Operaţii aritmetice

ARCCOS n	furnizează valoarea, în grade, a arccosinusului de n.
ARCCOT n	furnizează valoarea, în grade, a arccotangentei de n.
ARCSIN n	furnizează valoarea, în grade, a arcsinusului de n.

ARCTAN n	furnizează valoarea, în grade, a arctangentei de n.
COSINE n **COS** n	furnizează valoarea, în grade, a cosinusului de n.
COTANGENT n **COT** n	furnizează valoarea, în grade, a cotangentei de n.
DIV a b	furnizează cîtul obţinut prin împărţirea lui a prin b.
INT n	furnizează partea ÎNTreagă a lui n.
PRODUCT a b (**PRODUCT** a b ... n) **PROD** a b (**PROD** a b ... n)	furnizează produsul subiectelor (intrărilor). Este echivalentă cu operaţia de înmulţire simbolizată prin semnul ×.
RANDOM n	furnizează un număr aleator întreg, cuprins între 0 şi (n—1).

Notă: adunarea, scăderea, înmulţirea şi împărţirea pot fi utilizate în forma infix, în care operatorii (+ — × /) se pun între parametri de intrare.
Adunarea, înmulţirea şi împărţirea pot fi utilizate şi în forma prefix, dar în acest caz, SUM, PRODUCT, respectiv DIV, vor fi urmate de doi parametri de intrare. În LOGO se pot efectua operaţii aritmetice atît cu numere întregi, cît şi zecimale.

REMAINDER a b **REST** a b	furnizează restul împărţirii lui a prin b
ROUND n	furnizează n rotunjit la cel mai apropiat întreg.
SINE n **SIN** n	furnizează valoarea, în grade, a sinusului de n.
SQRT n	furnizează rădăcina pătrată din n; n trebuie să fie pozitiv.

SUM a b **SUM** a b ... n	furnizează suma subiectelor (intrărilor). Este echivalentă cu operaţia de adunare simbolizată prin semnul +.
TANGENT n **TAN** n	furnizează valoarea, în grade, a tangentei de n.
a + b	furnizează suma lui a cu b.

Notă: în operaţiile aritmetice (cu semne) se recomandă ca semnul să fie încadrat de spaţii, altfel **LOGO** poate interpreta că unul dintre subiecte este un număr negativ, de exemplu în cazul scăderii.

a — b	furnizează diferenţa dintre a şi b.
a × b	furnizează produsul dintre a şi b.
a/b	furnizează cîtul dintre a şi b (a împărţit la b).
a < b	furnizează valoarea logică TRUE dacă a este mai mic decît b şi FALSE în caz contrar.
a > b	furnizează valoarea logică TRUE dacă a este mai mare decît b şi FALSE în caz contrar.
a = b	furnizează TRUE dacă a este egal cu b, de oricare tip ar fi aceste subiecte (cuvinte, liste sau numere) şi FALSE în caz contrar.

5. Definiri şi editări de proceduri

EDIT procedură **ED** procedură	intrarea în modul de lucru de editare a procedurii.
EDNS nume	permite editarea numelor şi a va-

	lorilor lor.
EDNS [nume lista]	Fără subiect se vor lista toate variabilele cu valorile lor; cu subiect se vor lista numai variabilele numite, cu valorile lor.
END	indică sfîrşitul definirii unei proceduri.
TO nume	permite începerea definirii unei proceduri.

6. **Condiţii de control**

BYE	ieşire din LOGO cu trecerea controlului în interpretorul BASIC. Se poate intra din nou în LOGO prin comanda RUN.
IF condiţie [lista 1 de comenzi] [lista 2 de comenzi] **DACĂ** condiţie [lista 1 de comenzi] [lista 2 de comenzi]	dacă este îndeplinită condiţia, se execută lista 1 de instrucţiuni, dacă nu, se execută lista 2.
Formă particulară: **IF** condiţie [lista de instrucţiuni]	lista este executată doar dacă este îndeplinită condiţia; în caz contrar lista nu se execută şi se trece la linia LOGO care urmează.
OUTPUT obiect **OP** obiect	ieşire dintr-o procedură cu furnizarea unui rezultat, care se atribuie ca valoare numelui procedurii. **Notă:** o procedură se transformă în „funcţie" dacă se termină cu instrucţiunea de „ieşire" OUTPUT.
REPEAT n [listă de comenzi] **REPETĂ** n [listă de comenzi]	repetă de n ori comenzile din listă. **Notă:** n trebuie să fie pozitiv.

RUN [listă de comenzi] se execută lista ca o linie LOGO.

STOP oprește execuția procedurii curente și redă controlul procedurii care a apelat-o.

TOPLEVEL oprește execuția procedurii curente și redă controlul lui LOGO (apare promptul ?). Este echivalentă cu acționarea tastelor **CS** și **SPACE.**

7. **Operații logice**

Notă: Parametri de intrare pentru primitivele care reprezintă operații logice pot fi TRUE sau FALSE sau condiții. Termenul de predicat (pred) este folosit pentru a descrie o funcție ai cărei parametri de intrare pot fi TRUE sau FALSE sau condiții și care furnizează ca rezultat TRUE sau FLASE, cum este cazul operațiilor logice.

AND pred1 pred2
(**AND** pred1
pred2 ... predn)
SI pred1 pred2
SI pred1
pred2 ... predn)

furnizează TRUE dacă toți parametrii de intrare sînt TRUE, și FALSE în caz contrar.

NOT pred
Nu pred

furnizează TRUE dacă predicatul (parametrul de intrare) este FALSE, și FALSE dacă predicatul este TRUE.

OR pred1 pred2
(**OR** pred1
pred2 ... predn)
SAU pred1 pred2
(**SAU** pred1
pred2 ... predn)

furnizează TRUE dacă cel puțin unul din predicate este TRUE, și FALSE dacă nici unul din predicate nu este TRUE.

8. **Extragere rezultate**

KEYP furnizează TRUE dacă este acționată o anumită tastă ori o combinație de anumite taste valide, sau FALSE în caz contrar.

PRINT obiect **PR** obiect **SCRIE** obiect	afişează conţinutul obiectului (cuvînt, listă sau variabilă). Cuvîntul se va afişa fără ghilimele, iar lista fără paranteze pătrate.
READCHAR **RC** **CITCAR**	comandă calculatorului să aştepte apăsarea unei taste şi, apoi, furnizează caracterul introdus, care va fi preluat de o procedură, ca orice rezultat al unei operaţii.
READLIST **RL** **CITLIST**	furnizează lista dată de utilizator la tastatură. Întreaga linie, introdusă înainte de acţionarea tastei CR, este considerată o listă. În urma acţionării tastelor, caracterele corespunzătoare sînt afişate pe ecran.
SHOW obiect	afişează cuvîntul, lista sau numerele date ca parametri de intrare. Listele sînt afişate cu paranteze pătrate.
SOUND [durată înălţime] **SUNET** [durată înălţime]	produce un sunet de o anumită durată şi de o anumită înălţime. Durata sunetului se dă în secunde (de la 0 la 255), iar înălţimea în semitonuri. Do-ul central este reprezentat prin valoarea 0, apoi sunetele cu înălţimea mai mare prin numere întregi pozitive iar sunetele cu înălţimea mai mică prin numere întregi negative. Al doilea parametru trebuie cuprins între —62 şi 75.
TYPE obiect (**TYPE** obiect1 obiect2 ... obiectn)	afişează obiectele. Spre deosebire de PRINT şi SHOW, nu provoacă linefeed după afişare.
WAIT n **AŞTEAPTĂ** n	aşteaptă n/50 secunde.

9. Ecranul

Notă: după încărcarea lui LOGO, ecranul se află în modul text, putîndu-se afişa 22 linii de text. În modul grafic (în care se intră automat după orice comandă grafică), cele 22 de linii pot fi folosite pentru grafică, iar alte două linii, din partea de jos a ecranului, pentru mesaje.
Pe fiecare linie se pot afişa 32 de caractere. Pe prima coloană în modul comandă şi în modul de definire a unei proceduri este un prompt (care indică aşteptarea introducerii unei comenzi), iar ultima coloană este rezervată pentru semnul exclamării, care indică o linie LOGO neterminată, mai lungă de 32 de caractere.

BRIGHT n — stabileşte dacă afişarea va fi cu luminozitate (n = 1) sau nu (n = 0).

CLEARTEXT
CT
ŞTERGETEXT — şterge tot textul de pe ecran. În modul grafic, această comandă va ştege textul de pe ultimele două linii ale ecranului.

COPYSCREEN — realizează o copie a ecranului la imprimantă (hard copy).

CURSOR — furnizează o listă formata din două numere: primul va reprezenta coloana, iar al doilea linia pe care se găseşte cursorul.

FLASH — afişarea se va face pe un fond „clipitor" pînă la întîlnirea comenzii NORMAL.

INVERSE — afişarea se va face inversîndu-se culoarea pentru text şi desene cu cea a fondului pînă la întîlnirea comenzii NORMAL.

NORMAL — afişarea se va face fără „clipire" şi fără inversarea culorilor.

OVER n — afişarea se va face peste orice de-

	sen sau text (supratipărire) în cazul lui OVER 1. Cu OVER 0 se revine în modul obişnuit.
SETCURSOR [a b] **SETCUR** [a b] **FIXCURSOR** [a b]	mutarea cursorului în poziţia indicată de cei doi parametri din listă. Primul parametru va reprezenta coloana, iar al doilea linia. Coloanele sînt numerotate de la 0 la 30, iar liniile de la 0 la 21.
SETTC [a b]	specifică culoarea fondului şi a cernelii la afişarea unui text. Cei doi parametri pot lua valorile pentru culori (0 — 7).
TEXTCOLOUR **TC**	furnizează o listă de doi parametri, din care primul va specifica culoarea fondului şi al doilea culoarea cernelii la afişarea unui text.
TEXTSCREEN **TS**	trecerea din modul grafic al ecranului în modul text (întreg ecranul pentru text).

11. **Spaţiul de lucru**

Notă: o parte din memoria, iniţial disponibilă, fiind ocupată chiar de interpretorul LOGO, partea de memorie care rămîne la dispoziţia utilizatorului se numeşte **spaţiul de lucru.**

ERALL	şterge tot ce a fost creat (proceduri şi variabile) în spaţiul de lucru (ERases ALL). **Notă:** folosirea acestei comenzi va avea ca efect şi ştergerea tuturor procedurilor în limba română, acestea fiind şi ele memorate în spaţiul de lucru. Comanda ERALL nu afectează conţinutul curent al editorului. Pentru a şterge şi editorul se foloseşte ERALL şi EDIT [].

ERASE nume **ER** nume **UITA** nume	şterge procedura sau procedurile (din listă) din spaţiul de lucru.
ERN nume	şterge variabila (variabilele) numite din spaţiul de lucru (ERases Named).
ERNS	şterge numele şi valorile atribuite tuturor variabilelor din spaţiul de lucru (ERases NameS).
ERPS	şterge toate procedurile din spaţiul de lucru (ERases all the ProcedureS). **Notă:** folosirea acestei comenzi va avea ca efect şi ştergerea tuturor procedurilor in limba română.
PO nume	afişează definiţia procedurii sau procedurilor din listă (Prints Out).
POALL	afişează numele şi definiţiile tuturor procedurilor, precum şi valorile tuturor variabilelor din spaţiul de lucru (Print Out ALL).
PONS	afişează numele şi valorile tuturor variabilelor din spaţiul de lucru (Prints Out the NameS).
POPS	afişează definiţiile tuturor procedurilor curente din spaţiul de lucru (Prints Out ProcedureS).
POTS	afişează numele tuturor procedurilor definite (Prints Out the TitleS).

12. Salvări și încărcări de fișiere

Notă: toate procedurile definite în timpul unei sesiuni de lucru sînt memorate de către LOGO în spațiul de lucru. Toate realizările (proceduri, desene, etc) pot fi salvate oricînd pe caseta magnetică și refolosite, la nevoie, prin încărcarea lor în memoria calculatorului de pe același suport magnetic. Realizările pot fi aranjate în fișiere pe trei tipuri:

a) fișierul de proceduri LOGO (tipul LOG) este un fișier care conține proceduri LOGO. De exemplu, ROMANA sau DESENE de pe caseta LOGO sînt fișiere de proceduri LOGO.

b) fișierul editor (tipul TXT) este reprezentat prin conținutul curent al editorului LOGO.

c) fișierul ecran (tipul SCR) este reprezentat prin imaginea ecran curentă.

LOAD " nume fișier " tip fișier	încarcă fișierul **nume fișier** de pe casetă în spațiul de lucru. Dacă extensia care precizează tipul fișierului lipsește, atunci se consideră implicit că tipul este LOG. În timpul încărcării sînt afișate pe ecran numele procedurilor din fișier, iar după încărcare, promptul LOGO (?) va reapărea pe ecran.
LOAD " nume fișier "	încarcă fișierul editor salvat cu SAVED " nume fișier și îl face conținutul curent al editorului.
LOADSCR " nume fișier "	încarcă fișierul ecran salvat cu SAVESCR" nume_fișier și afișează conținutul său (imaginea) pe ecran.
PRINTON	tipărește la imprimantă tot ce se va afișa pe ecran.
PRINTOFF	oprirea tipăririi la imprimantă.

SAVE " nume fişier» [nume] — salvează procedurile din lista de proceduri sub forma unui fişier cu numele nume fişier. Numele dat fişierului poate avea pînă la 7 caractere.

SAVEALL " nume fişier» — salvează tot ce există în spaţiul de lucru (proceduri şi variabile) sub numele nume fişier.

SAVED " nume fişier» — salvează tot ce există în Editorul LOGO sub numele nume fişier.

SAVESCR " nume fişier» — salvează imaginea ecran sub numele nume fişier.

13. Funcţii şi primitive avansate

Notă: unele primitive permit ca procedurile să fie definite şi mînuite din interiorul altor proceduri.
Unele primitive afectează însuşi sistemul LOGO. Acestea se pot utiliza în vederea citirii conţinutului memoriei. Se recomandă însă salvarea în prealabil a spaţiului de memorie, acesta putîndu-se altfel distruge (în mod accidental). În general, numele acestor primitive începe cu un punct.

NODES — furnizează numărul de noduri libere sau, cu alte cuvinte, spaţiul de memorie disponibil în spaţiul de lucru pentru proceduri, variabile şi rulări de proceduri. Un nod ocupă 5 octeţi. Prin utilizarea lui NODES, imediat după RECYCLE, se furnizează numărul de noduri care sînt încă libere.

RECYCLE	eliberează maximum posibil de noduri prin procedeul „colectarea gunoiului".
COPYDEF "nume nou" nume	copiază definiţia unui nume de procedură existentă sub un alt nume (nou). Prcedura existentă nu este ştearsă.
DEFINE " nume" [param ... paramn] [liniel ... linien]	Comanda are doi parametri de intrare: primul este numele unei proceduri, iar al doilea o listă. Elementele listei sînt: o listă de parametri de intrare pentru noua procedură şi, al doilea, o listă ale cărei elemente sînt liniile procedurii. Cu această comandă se pot scrie proceduri care definesc alte proceduri.
● **CONTENTS**	furnizează o listă care include toate procedurile, variabile, etc, din sistemul LOGO.
● **PRIMITIVES**	listează toate primitivele LOGO.
● **RESERVE** n	rezervă n octeţi în spaţiul de lucru pentru un program în cod maşină. Poate fi utilizată numai la începutul unei sesiuni de lucru LOGO.
● **RESERVED**	furnizează adresa de început şi sfîrşit a zonei de memorie rezervată prin RESERVE.
● **BLOAD** " nume fişie " adresă	încarcă în memorie un fişier de pe suport extern, începînd de la adresa specificată.

- **BSAVE** "nume fişier" [adresă n] — salvează n octeţi de la adresa specificată, sub un fişier al cărui nume este indicat.
- **SETSERIAL** n — stabileşte viteza de transmisie. n poate fi 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200. Iniţial, viteza de transmisie este 9600 bauds.
- **SERIALIN** — citeşte un octet de la portul serial (Interfaţa RS232).
- **SERIALOUT** n — trimite un octet la portul serial (interfaţa RS232).
- **DEPOSIT** adresă n — plasează valoarea **n** în locaţia de memorie specificată prin adresă. Poate fi utilizată pentru a defini un caracter grafic utilizator (UDG).
- **EXAMINE** adresă — regăseşte o valoare memorată la adresa specificată.
- **CALL** adresă — apelează un program în cod maşină, instalat în prealabil printr-o comandă „BLOAD".

Jocuri realizate de Cristian Constantinescu şi Ion Diamandi

RECOOP a realizat următoarele programe pentru calculator:

1. — **JOCURI**: Şah, Jump (salt), Rubic (Jocul pătratelor), GO (Gomoku), Mastermind, Grafice (joc matematic), Lab (culesul cireşelor), Reversi, MLS, Rebec, Dipo (Vulpi şi iepuri) **(Caseta nr. 1)**

2. — **JOCURI**: GO (Gomoku), Mastermind, Şah, Reversi, Lab (Labirint) **(Caseta nr. 2)**

3. — **JOCURI**: Fotbal logic, Superevol, Raliu, Simultan **(Caseta nr. 3)**

4. — **JOCURI**: Logic IM-2, Biliard, Iepure, Simultan **(Caseta nr. 4)**

5. — **JOCURI**: Tictactoe, Broscuţe, Deplasare, Nim, Cărare, Drum, Pătrate, Impas, Triplet, A (Animale), Obstacole, Jungla, Traversare, Vrăjitorul, Ariadna, Vînătoare, Robac, Domino, Hang, Hanoi, Loyd, Predau, Peştera, Labirint, Dipo, Şeptică, Dist. **Caseta nr. 5)**

6. — **CHIMIE**: Acizi, Formula, Mendeleev, Valenţă, Ecuaţii **(Caseta nr. 6)**

7. — **JOCURI**: Logo, Română, Desene, Joc, Şah, Titato, Foto

(Caseta nr. 7)

8. — **LIMBA STRĂINĂ**: Limba engleză, Hang, Cuvintele vrăjite, Rebus

(Caseta nr. 8)

9 — **FIZICA**: În curs de apariţie

(Caseta nr. 9)

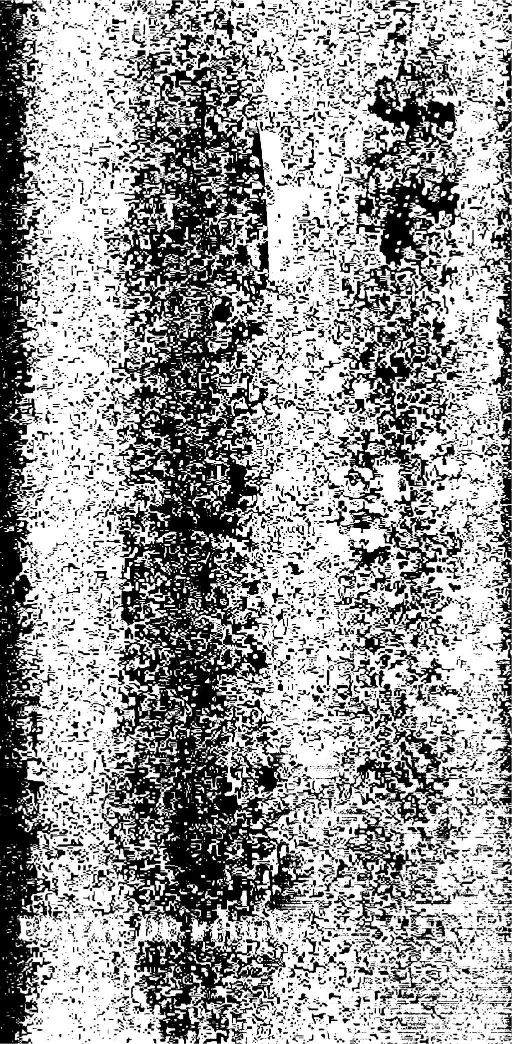
10. — **MATEMATICA**: Izometrii, TLS (Tangenta, limită a secantei), Triunghi

(Caseta nr. 10)

NOTĂ: Cei care pot elabora programe pentru jocuri, diferite discipline de învăţămînt, grafică etc. şi doresc să colaboreze cu RECOOP, se pot adresa la tel. 13 81 75, 15 04 10, 15 72 93 / interior 112 sau 115.

Tiparul executat la I. P. Sibiu

Caseta nr. 7



LEI 3,50